

Application No. 10/827117 (Docket: NEXTIO.0402)
37 CFR 1.111 Amendment dated 08/08/2006
Reply to Office Action of 05/08/2006

AMENDMENTS TO THE SPECIFICATION

Please replace the TITLE with the following amended TITLE:

~~SWITCHING APPARATUS AND METHOD FOR
PROVIDING SHARED I/O WITHIN A LOAD-STORE FABRIC~~
APPARATUS AND
METHOD FOR SHARING I/O ENDPOINTS WITHIN A LOAD STORE FABRIC BY
ENCAPSULATION OF DOMAIN INFORMATION IN TRANSACTION LAYER
PACKETS

Please replace paragraph [0005] with the following amended paragraph:

[0005] This application is related to the following co-pending U.S. Patent Applications, which have a common assignee and common inventors.

<u>SERIAL NUMBER</u>	<u>FILING DATE</u>	<u>TITLE</u>
<u>10/827622</u> (NEXTIO.0400)	4/19/2004	SWITCHING APPARATUS AND METHOD FOR PROVIDING SHARED IO WITHIN A LOAD-STORE FABRIC
<u>10/827620</u> (NEXTIO.0401)	4/19/2004	SWITCHING APPARATUS AND METHOD FOR PROVIDING SHARED IO WITHIN A LOAD-STORE FABRIC

Please delete the section entitled "SUMMARY OF THE INVENTION" in its entirety and substitute the following section therefor:

SUMMARY OF THE INVENTION

[0028] The present invention, among other applications, is directed to solving the above-noted problems and addresses other problems, disadvantages, and limitations of the prior art. The present invention provides a superior technique for sharing I/O endpoints within a load-store infrastructure. In one embodiment, a switching apparatus for sharing input/output (I/O) endpoints is provided. The switching apparatus has a first plurality of I/O ports, a second I/O port, and core logic. The first plurality of I/O ports is coupled to a plurality of operating system domains (OSDs) through a load-store fabric, each routing transactions between the plurality of OSDs and the switching apparatus. The second I/O

Application No. 10/827117 (Docket: NEXTIO.0402)
37 CFR 1.111 Amendment dated 08/08/2006
Reply to Office Action of 05/08/2006

port is coupled to a first shared input/output endpoint. The first shared input/output endpoint requests/completes the transactions for each of the plurality of OSDs. The core logic is coupled to the first plurality of I/O ports and the second I/O port. The core logic routes the transactions between the first plurality of I/O ports and the second I/O port. The core logic designates a corresponding one of the plurality of OSDs according to a variant of a protocol, where the variant includes encapsulating an OS domain header within a transaction layer packet that otherwise comports with the protocol, and where the protocol provides for routing of the transactions only for a single OSD.

[0029] One aspect of the present invention contemplates a shared input/output (I/O) switching mechanism. The shared I/O switching mechanism has core logic that enables operating system domains to share one or more I/O endpoints over a load-store fabric. The core logic includes global routing logic. The global routing logic routes first transactions to/from said operating system domains, and routes second transactions to/from the one or more I/O endpoints. Each of the second transactions designates an associated one of the operating system domains for which an operation specified by each of the first transactions be performed. The associated one of the operating system domains is designated according to a variant of a protocol, where the variant includes encapsulating an OS domain header within a transaction layer packet that otherwise comports with the protocol, and where the protocol provides exclusively a single operating system domain within the load-store fabric.

[0030] Another aspect of the present invention comprehends a method for interconnecting independent operating system domains to a shared I/O endpoint within a load-store fabric. The method includes: via first ports, first communicating with each of the independent operating system domains according to a protocol that provides exclusively for a single operating system domain within the load-store fabric; via a second port, second communicating with the shared I/O endpoint according to a variant of the protocol to enable the shared I/O endpoint to associate a prescribed operation with a corresponding one of the independent operating system domains. The second communicating includes encapsulating an OS domain header within a transaction layer packet that otherwise comports with the protocol, where the value of the OS domain

Application No. 10/827117 (Docket: NEXTIO.0402)
37 CFR 1.111 Amendment dated 08/08/2006
Reply to Office Action of 05/08/2006

header designates the corresponding one of the operating system domains. The method also includes: via core logic within a switching apparatus, mapping the independent operating system domains to the shared I/O endpoint.